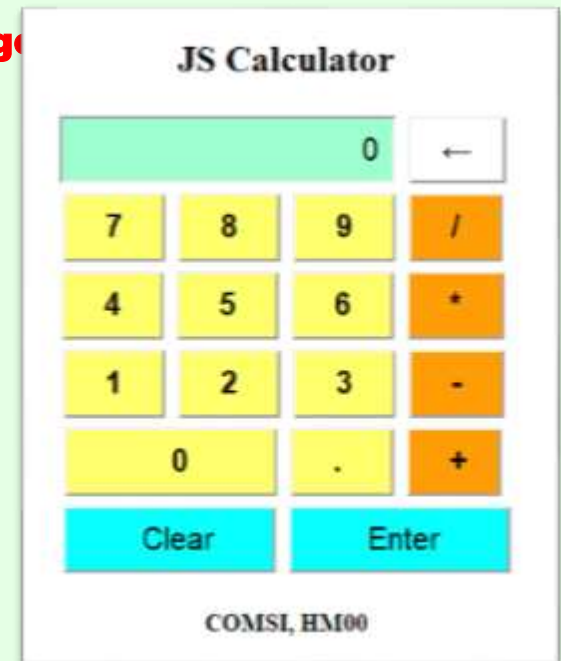


# HTML5



# Weekly plan (HTML5, 1<sup>st</sup> semester 2022)

- **wk01** : Introduction to curriculum, current state of HTML5 & github
- **wk02** : Making HTML5 documents
- **wk03** : iFrame, Media, my 1<sup>st</sup> Homepage & Intro to Semantic tags
- **wk04** : **Semantic tags & Web forms**
- **wk05** : **CSS3 I. Basic & Box model**
- **wk06** : **CSS3 II. Advanced & Animation**
- **wk07** : DIY:x-mas-card , **Quiz-15, my 2<sup>nd</sup> Homepage**
- **wk08** : Mid-term Exam. **my 2<sup>nd</sup> Homepage**
- **wk09** : **JS I. Data types & operators**
- **wk10** : **JS II. if, loop & functions**
- **wk11** : **JS III.**
- **wk12** : **JS IV.**
- **wk13** : **JS V.**
- **wk14** : **my final Homepage (hmn\_rpt03)**
- **wk15** : Final exam.



# My ID (HTML5, 1<sup>st</sup> semester 2022)

ID	성명
HM01	김정현
HM02	오세현
HM03	김기덕
HM04	강대진
HM05	김성우
HM06	김창연
HM07	김창욱
HM08	김태화
HM09	박세훈
HM10	박신영

ID	성명
HM11	박제홍
HM12	이승무
HM13	이승준
HM14	이재하
HM15	이준희
HM16	이현준
HM17	임태형
HM18	정동현
HM19	정희서
HM20	김동영
HM21	정희철
HM22	조동현

# JS-project-I. hmxx\_calc.html

4

[JS project I.] 자바스크립트를 이용하는 간단한 계산기를 완성하시오.

- ① 계산기 - 다음 연산 구현 : add(+), subtract(-), **multiply(\*), divide(/), mod(%)**
- ② 파일명: **hmxx\_calc.html**
- ③ 가점: Javascript 프로그래밍 응용 능력.

```
function calc_mod() {  
    // 나눗셈  
    var x = document.getElementById("x").value;  
    var y = document.getElementById("y").value;  
    var result;  
    result = parseInt(x) % parseInt(y);  
    document.getElementById("result").value = result;  
}
```

# wk09-실습 : 결과를 나의 github에 올리기

5

실습 결과를 github에 올립니다.

1. README.md에는 실습 결과 요약 추가 입력
2. hmxx\_calc.html 완성
3. "hmxx" repo 에 wk09 폴더 upload

단 업로드가 안될 경우, wk09.zip을 업로드  
그리고 집에서 wk09 폴더로 다시 업로드.



# 06

자바스크립트 언어

# 강의 목표

1. 자바스크립트 언어의 요소와 구조를 이해한다.
2. 자바스크립트 코드를 웹 페이지에 삽입하는 방법을 안다.
3. 자바스크립트로 브라우저에 출력하고 사용자 입력 받는 방법을 안다.
4. 자바스크립트에서 다루는 데이터 타입과 변수에 대해 이해한다.
5. 자바스크립트의 연산자의 종류를 알고 사용할 수 있다.
6. 자바스크립트의 조건문의 종류를 알고 사용할 수 있다.
7. 자바스크립트의 반복문의 종류를 알고 사용할 수 있다.
8. 자바스크립트 함수를 작성할 수 있다.
9. 사용자에게 제공되는 `eval()`, `parseInt()` 등 자바스크립트 함수를 활용할 수 있다.

→ JS project II. Fancy JS calculator 제작

# 자바스크립트 식과 연산 II.

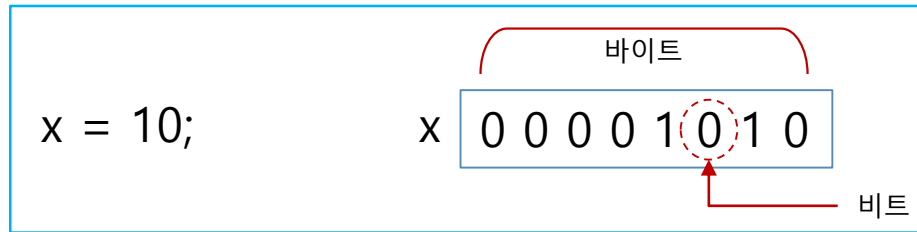
연산 종류	연산자	연산 종류	연산자
산술	+ - * / %	대입	= *= /= += -= &= ^=  = <<= >>= >>>=
증감	++ --	비교	> < >= <= == !=
비트	&   ^ ~	논리	&&    !
시프트	>> << >>>	조건	? :



# 비트 연산 ( &, |, ^, ~ )

9

## □ 비트 개념



## □ 비트 연산 종류

- ▣ 비트들끼리의 비트 논리 연산
- ▣ 비트 시프트 연산

# 비트 논리 연산

10

## □ 비트 논리 연산

연산자	별칭	연산 설명
$a \& b$	비트 AND 연산	두 비트 모두 1이면 1, 그렇지 않으면 0
$a   b$	비트 OR 연산	두 비트 모두 0이면 0, 그렇지 않으면 1
$a \wedge b$	비트 XOR 연산	두 비트가 다르면 1, 같으면 0
$\sim a$	비트 NOT 연산	1을 0으로, 0을 1로 변환

$a = 106;$  01101010

$b = 77;$  01001101

$c = a \& b;$

0	1	1	0	1	0	1	0
&	0	1	0	0	1	1	0
<hr/>							
c	0	1	0	0	1	0	0

둘 다 1,  
결과 1

하나라도 0,  
결과 0

$c = a | b;$

0	1	1	0	1	0	1	0
	0	1	0	0	1	1	0
<hr/>							
c	0	1	1	0	1	1	1

둘 다 0,  
결과 1

하나라도 1,  
결과 1

$c = a \wedge b;$

0	1	1	0	1	0	1	0
^	0	1	0	0	1	1	0
<hr/>							
c	0	0	1	0	0	1	1

둘이 같으면  
결과 0

둘이 다르면,  
결과 1

$c = \sim a;$

~	0	1	1	0	1	0	1
<hr/>							
c	1	0	0	1	0	1	0

1은 0으로  
바꿈

0은 1로  
바꿈

# 비트 시프트 연산

11

- 시프트 : 저장 공간에서 비트들의 오른쪽/왼쪽 이동



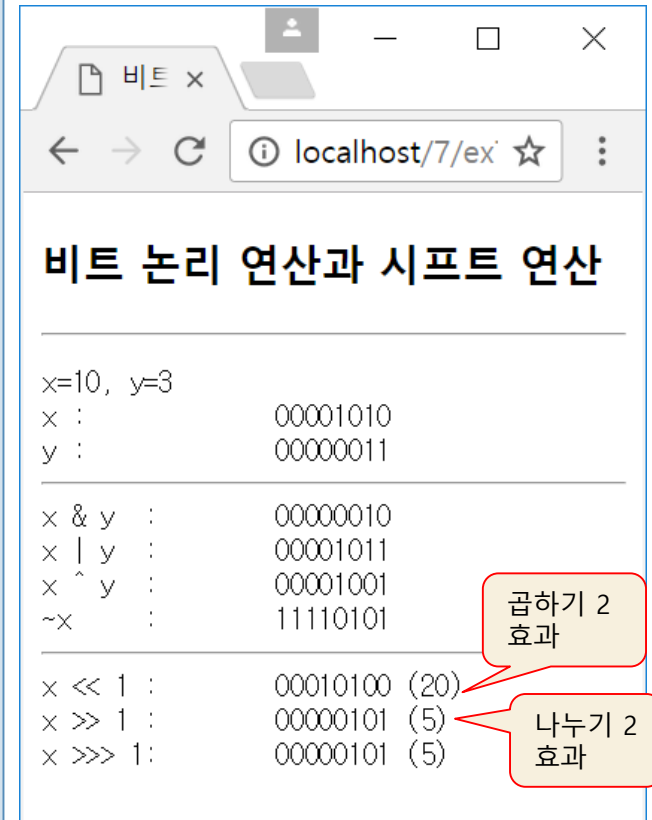
연산자	별칭	설명
<code>a &lt;&lt; b</code>	산술적 왼쪽 시프트	a의 비트들을 왼쪽으로 b번 이동. 최하위 비트의 빈자리는 0으로 채움. 한 비트 시프트마다 곱하기 2의 효과 발생. a 값은 변화 없음
<code>a &gt;&gt; b</code>	산술적 오른쪽 시프트	a의 비트들을 오른쪽으로 b번 이동. 최상위 비트의 빈자리는 시프트 전 최상위 비트로 채움. 한 비트 시프트마다 나누기 2의 효과 발생. a 값은 변화 없음
<code>a &gt;&gt;&gt; b</code>	논리적 오른쪽 시프트	a의 비트들을 오른쪽으로 b번 이동. 최상위 비트의 빈자리는 0으로 채움. a 값은 변화 없음

# 예제 6-13 비트 연산

12

```
<!DOCTYPE html>
<html> <head> <title>비트 연산</title>
<script>
function digit8(v) { // 숫자 v를 8비트 2진수로 변환
    var str="";
    for(i=0; i<8; i++, v<=<=1) {
        if((v & 0x80)) str += "1";
        else str += "0";
    }
    return str;
}
</script>
</head>
<body>
<h3>비트 논리 연산과 시프트 연산</h3>
<hr>
<script>
    var x=10, y=3;
    document.write("<pre>");
    document.write("x=" + x + ", y=" + y + "<br>");
    document.write("x :      " + digit8(x) + "<br>");
    document.write("y :      " + digit8(y) + "<br>");
    document.write("<hr>");
    document.write("x & y :      " + digit8(x&y) + "<br>");
    document.write("x | y :      " + digit8(x|y) + "<br>");
    document.write("x ^ y :      " + digit8(x^y) + "<br>");
    document.write("~x :      " + digit8(~x) + "<br>");
    document.write("<hr>");
    document.write("x << 1 :      " + digit8(x<<1) + " (" + (x<<1) + ")<br>");
    document.write("x >> 1 :      " + digit8(x>>1) + " (" + (x>>1) + ")<br>");
    document.write("x >>> 1:      " + digit8(x>>>1) + " (" + (x>>>1) + ")");
    document.write("</pre>");
</script>
</body>
</html>
```

wk10js\_ex13\_bit\_operation.html



x 가 -10일 때는 어떻게 달라지는 가를 확인 하시오.

## 예제 6-13 비트 연산

13

```
<script>
function digit8(v) { // 숫자 v를 8비트 2진수로 변환
    var str="";
    for(i=0; i<8; i++, v<<=1) {
        if((v & 0x80)) str += "1";
        else str += "0";
    }
    return str;
}
</script>
```

자바스크립트  
문자열 연산

**str1 + str2**

# 문자열 연산자

15

## □ 문자열 연결

■ +, +=

```
"abc" + "de"      // "abcde"
"abc" + 23        // "abc23"
23 + "abc"        // "23abc"
23 + "35"         // "2335"
23 + 35           // 58, 정수 더하기
```

■ 순서에 유의

```
23 + 35 + "abc"; // 23 + 35 -> 58로 먼저 계산, 58 + "abc" -> "58abc"
```

```
"abc" + 23 + 35; // "abc" + 23 -> "abc23"로 먼저 계산, "abc23" + 35 -> "abc2335"
```

## □ 문자열 비교

■ 비교 연산자(!=, ==, >, <, <=, >=)는 문자열 비교에 사용

■ 사전 순으로 비교 결과 리턴

```
var name = "kitae";
var res = (name == "kitae"); // 비교 결과 true, res = true
```

```
var res = (name > "park"); // name이 "park"보다 사전순으로 앞에 나오므로 res = false
```

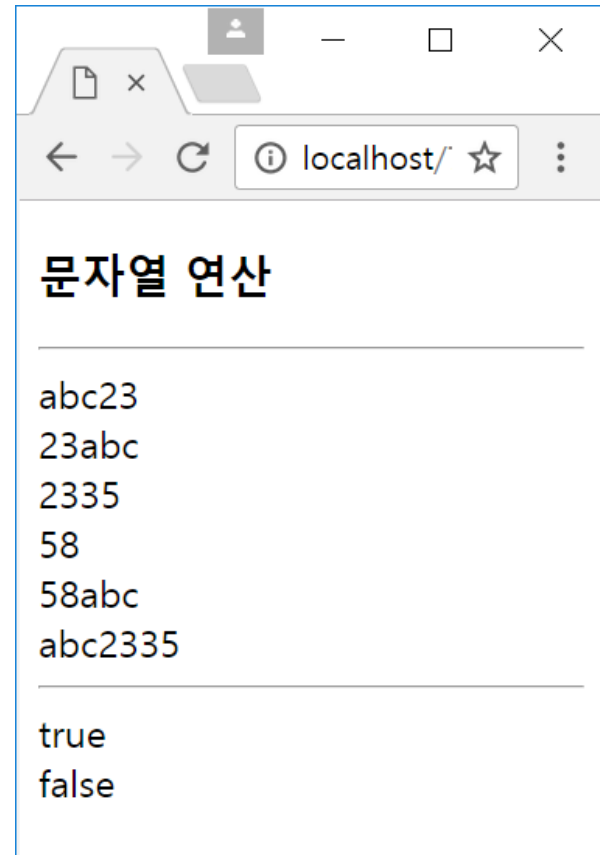


# 예제 6-14 문자열 연산

16

```
<!DOCTYPE html>
<html>
<head> <title> 문자열 연산 </title> </head>
<body>
<h3>문자열 연산</h3>
<hr>
<script>
  document.write("abc" + 23 + "<br>");
  document.write(23 + "abc" + "<br>");
  document.write(23 + "35" + "<br>");
  document.write(23 + 35 + "<br>");
  document.write(23 + 35 + "abc" + "<br>");
  document.write("abc" + 23 + 35 + "<br><hr>");

  var name = "kitae";
  document.write(name == "kitae");
  document.write("<br>");
  document.write(name > "park");
</script>
</body>
</html>
```



wk10js\_ex14\_string\_operation.html

# 자바스크립트 조건문

if

switch

# if, if-else

18

## □ if, if-else 문

```
if(조건식) {  
    ... 실행문 ... // 조건식이 참인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}
```

```
if(조건식) {  
    ... 실행문1 ... // 조건식이 참인 경우  
}  
else {  
    ... 실행문2 ... // 조건식이 거짓인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}  
else {  
    document.write("a가 크지 않다");  
}
```

```
if(조건식1) {  
    실행문1 // 조건식1이 참인 경우  
}  
else if(조건식2) {  
    실행문2 // 조건식2가 참인 경우  
}  
.....  
else {  
    실행문n; // 앞의 모든 조건이 거짓인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}  
else if(a < b) {  
    document.write("b가 크다");  
}  
else  
    document.write("a와 b는 같다");
```

# 예제 6-15 if-else 사용

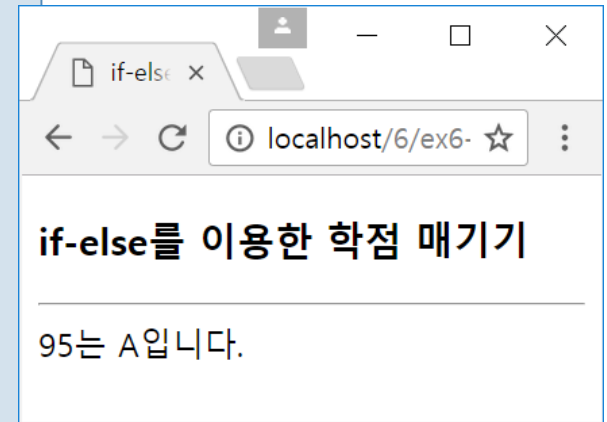
19

```
<!DOCTYPE html>
<html>
<head> <title> if-else </title> </head>
<body>
<h3>if-else를 이용한 학점 매기기</h3>
<hr>
<script>
  var grade;
  var score = prompt("황기태 님 점수를 입력하세요", 100);
  score = parseInt(score); // 문자열을 숫자로 바꿈
  if(score >= 90) // score가 90 이상
    grade = "A";
  else if(score >= 80) // 80 이상 90 미만
    grade = "B";
  else if(score >= 70) // 70 이상 80 미만
    grade = "C";
  else if(score >= 60) // 60 이상 70 미만
    grade = "D";
  else // 60 미만
    grade = "F";
  document.write(score + "는 " + grade + "입니다.<br>");
</script>
</body>
</html>
```

localhost 내용:

황기태 님 점수를 입력하세요

확인 취소



# switch 문

20

## □ switch 문

- ▣ 값에 따라 서로 다른 코드를 실행할 때, switch 문 적합

```
switch(식) {  
  case 값1: // 식의 결과가 값1과 같을 때  
    실행 문장 1;  
    break;  
  case 값2: // 식의 결과가 값2와 같을 때  
    실행 문장 2;  
    break;  
  ...  
  case 값m:  
    실행 문장 m; // 식의 결과가 값과 같을 때  
    break;  
  default: // 어느 값과도 같지 않을 때  
    실행 문장 n;  
}
```

```
var fruits="사과";  
switch(fruits) {  
  case "바나나":  
    price = 200; break;  
  case "사과":  
    price = 300; break;  
  case "체리":  
    price = 400; break;  
  default:  
    document.write("팔지 않습니다.");  
    price = 0;  
}
```

// switch 문의 실행 결과 price=300

# case 문의 '값'

21

- case 문의 '값'은 상수 (리터럴)만 가능

```
case 1 :  
case 2.7 :  
case "Seoul" :  
case true :
```

- case 문의 '값'에 변수나 식은 사용 불가

```
case a :           // 오류. 변수 a 사용 불가  
case a > 3:        // 오류. 식(a > 3) 사용 불가
```

# switch 문에서 break 문의 역할

22

## □ break 문

### ▣ switch 문 종료

- break; 문을 만날 때까지 아래로 코드 계속 실행

```
var city="Seoul";  
switch(city) {  
  case "Seoul":  
    document.write("서울");  
    break;  
  case "NewYork":  
    document.write("뉴욕");  
    break;  
  case "Paris":  
    document.write("파리");  
    break;  
}
```

서울뉴욕

(a) break;를 만날 때까지 아래로 실행을 계속하는 사례

```
var day="월";  
switch(day) {  
  case "월":  
  case "화":  
  case "수":  
  case "목":  
    document.write("정상영업");  
    break;  
  case "토":  
  case "일": document.write("휴일");  
    break;  
}
```

정상영업

(b) 여러 case에 대해 동일한 코드를 실행하도록 의도적으로 break;를 생략한 경우



# 예제 6-16 switch 문 사용

23

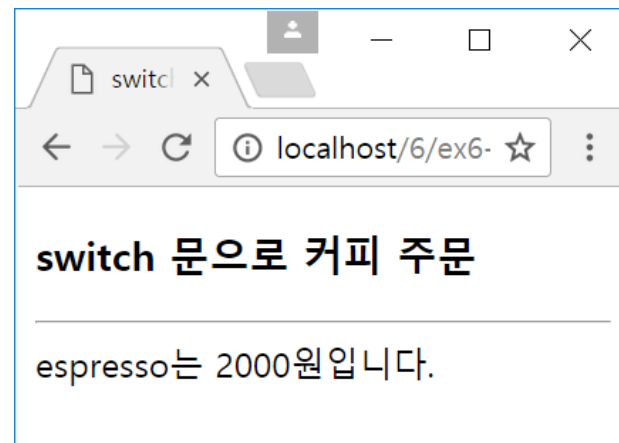
```
<!DOCTYPE html>
<html>
<head> <title>switch</title> </head>
<body>
<h3>switch 문으로 커피 주문</h3>
<hr>
<script>
  var price = 0;
  var coffee = prompt("무슨 커피 드릴까요?", "");
  switch(coffee) {
    case "espresso" :
    case "에스프레소" : price = 2000;
      break;
    case "카푸치노" : price = 3000;
      break;
    case "카페라떼" : price = 3500;
      break;
    default :
      document.write(coffee + "는 없습니다.");
  }
  if(price != 0)
    document.write(coffee + "는 " + price + "원입니다.");
</script>
</body>
</html>
```

"espresso"나  
"에스프레소"의 경우  
모두 실행

localhost 내용:

무슨 커피 드릴까요?

확인 취소



# 자바스크립트 반복문

**for**  
**while**  
**do while**

# 반복문

25

## □ for 문



```
// 0에서 9까지 출력  
for(var i=0; i<10; i++) {  
    document.write(i);  
}
```

0123456789

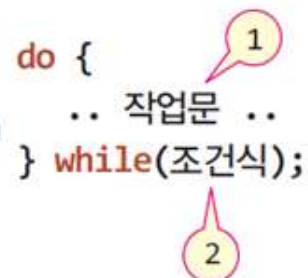
## □ while 문



```
var i=0;  
while(i<10) { // i가 0에서 9까지 반복  
    document.write(i);  
    i++;  
}
```

0123456789

## □ do-while 문



```
var i=0;  
do { // i가 0에서 9까지 반복  
    document.write(i);  
    i++;  
} while(i<10);
```

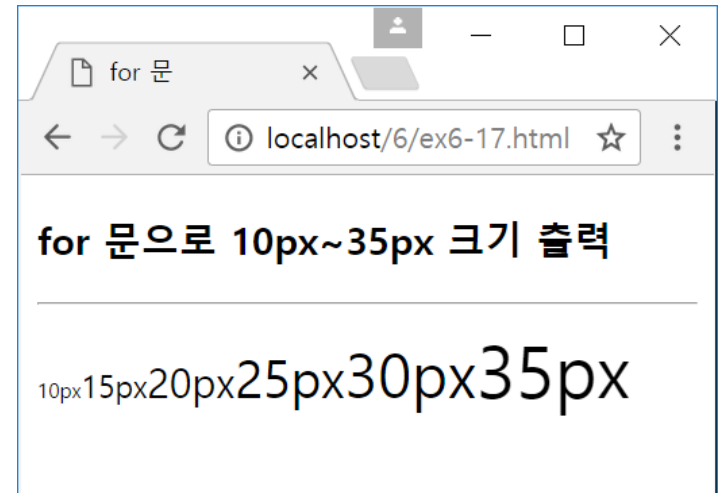
0123456789

# 예제 6-17 for 문으로 10px~35px 크기로 출력

26

```
<!DOCTYPE html>
<html>
<head>
<title>for 문</title>
</head>
<body>
<h3>for 문으로 10px~35px 크기 출력</h3>
<hr>
<script>
  for(var size=10; size<=35; size+=5) { // 5씩 증가
    document.write("<span ");
    document.write("style='font-size:" + size + "px'>");
    document.write(size + "px");
    document.write("</span>");
  }
</script>
</body>
</html>
```

wk10js\_ex17\_for.html



# 예제 6-18 while 문으로 0~n까지의 합 구하기

27

```
<!DOCTYPE html>
<html>
<head>
<title>while 문</title>
</head>
<body>
<h3>while 문으로 0에서 n까지 합</h3>
<hr>
<script>
  var n = prompt("0보다 큰 정수를 입력하세요", 0);
  n = parseInt(n); // 문자열 n을 숫자로 바꿈

  var i=0, sum=0;
  while(i<=n) { // i가 0에서 n까지 반복
    sum += i;
    i++;
  }
  document.write("0에서 " + n + "까지 합은 " + sum);
</script>
</body>
</html>
```

prompt()가 리턴한 것은 문자열

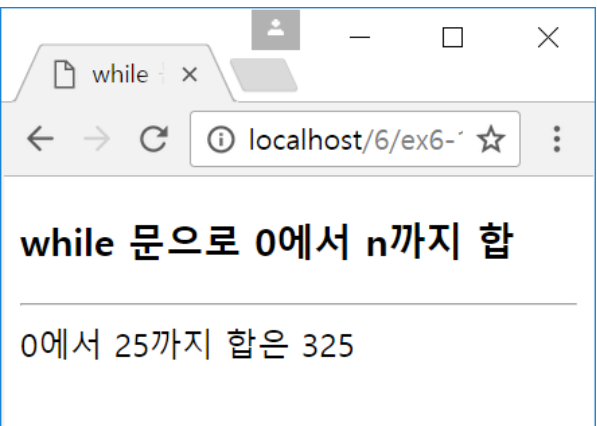
localhost 내용:

0보다 큰 정수를 입력하세요

25

확인

취소



wk10js\_ex18\_while.html

# 예제 6-19 do-while 문으로 0~n까지 합 구하기

28

```
<!DOCTYPE html>
<html>
<head>
<title>do-while 문</title>
</head>
<body>
<h3>do-while 문으로 0에서 n까지 합</h3>
<hr>
<script>
  var n = prompt("0보다 큰 정수를 입력하세요", 0);
  n = parseInt(n); // 문자열 n을 숫자로 바꿈

  var i=0, sum=0;
  do {
    sum += i;
    i++;
  } while(i<=n); // i가 0~n까지 반복
  document.write("0에서 " + n + "까지 합은 " + sum);
</script>
</body>
</html>
```

prompt()가 리턴한 것은 문자열

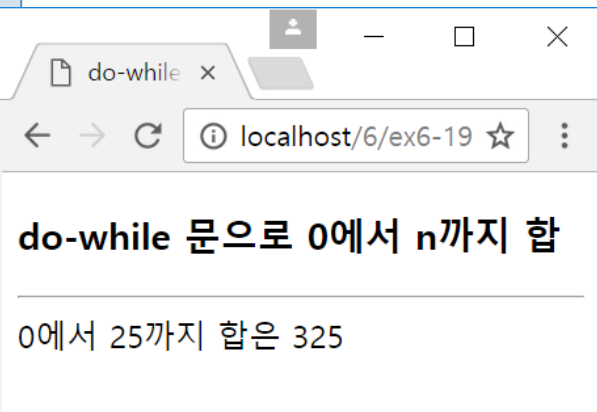
localhost 내용:

0보다 큰 정수를 입력하세요

25

확인

취소



wk10js\_ex19\_dowhile.html

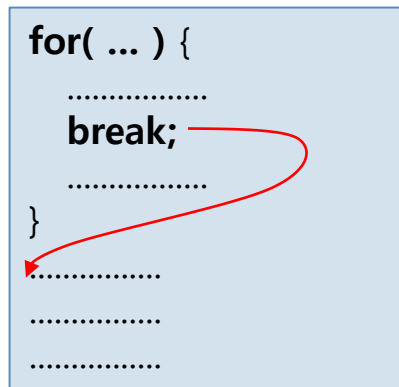
**break**  
**&**  
**continue**



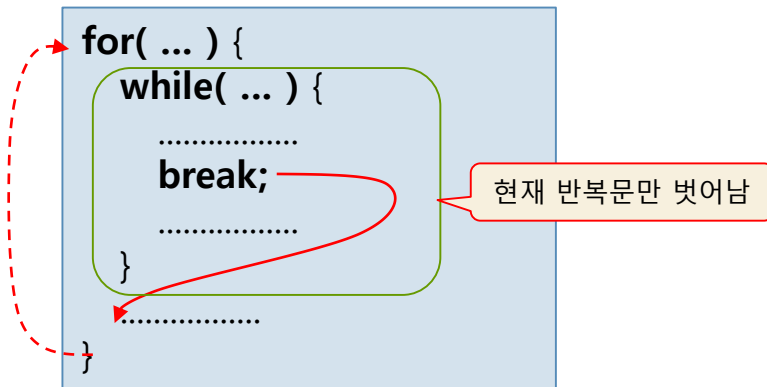
# 반복문 내의 break 문과 continue 문

30

- break 문 : 가장 안쪽 반복문 하나만 벗어나도록 제어

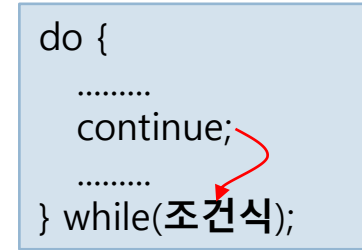
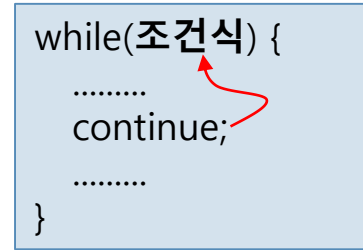
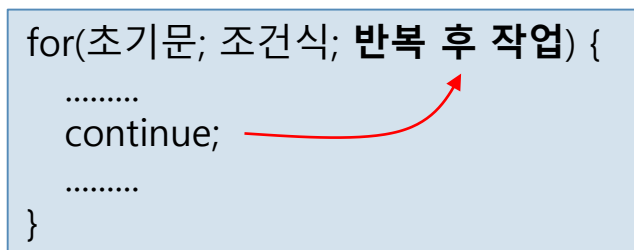


(a) 반복문 벗어나기



(b) 중첩 반복에서 현재 반복문만 벗어나

- continue 문 : 반복 코드 실행 중단, 다음 반복으로 점프

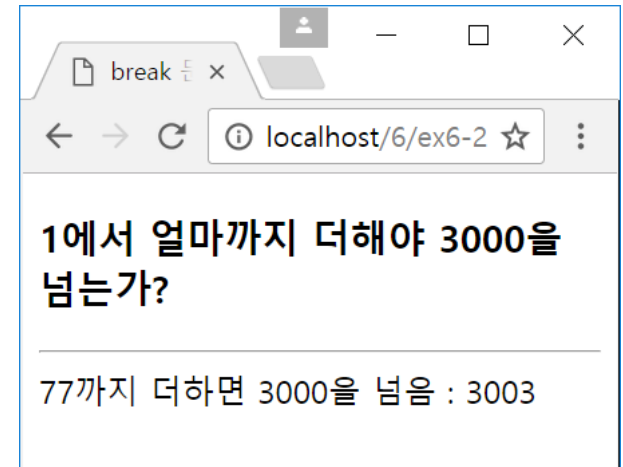


# 예제 6-20 break 문

31

```
<!DOCTYPE html>
<html>
<head>
<title>break 문</title>
</head>
<body>
<h3>1에서 얼마까지 더해야 3000을 넘는가?</h3>
<hr>
<script>
  var i=0, sum=0;
  while(true) { // 무한 반복
    sum += i;
    if(sum > 3000)
      break; // 합이 3000보다 큼. 반복문 벗어남
    i++;
  }
  document.write(i + "까지 더하면 3000을 넘음 : " + sum);
</script>
</body>
</html>
```

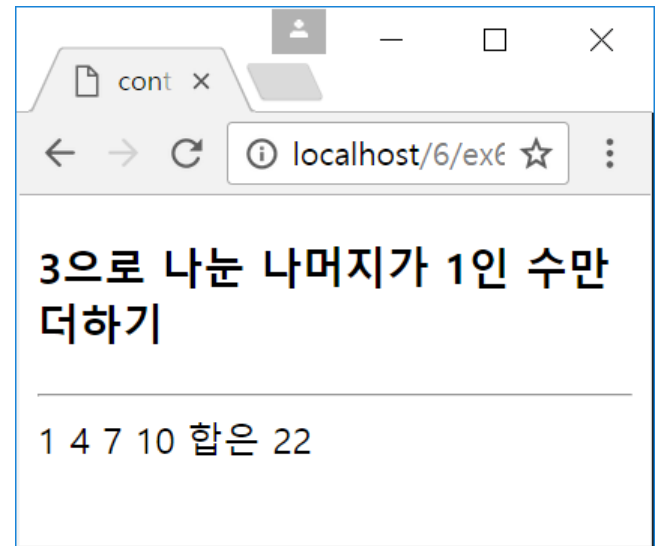
wk10js\_ex20\_break.html



# 예제 6-21 continue 문

32

```
<!DOCTYPE html>
<html>
<head>
<title>continue 문</title>
</head>
<body>
<h3>3으로 나눈 나머지가 1인 수만 더하기</h3>
<hr>
<script>
  var sum=0;
  for(i=1; i<=10; i++) { // i가 1에서 10까지 반복
    if(i%3 != 1) // 3으로 나눈 나머지가 1이 아닌 경우
      continue; // 다음 반복으로 점프(i++ 코드로)
    document.write(i + " ");
    sum += i;
  }
  document.write("합은 " + sum);
</script>
</body>
</html>
```



wk10js\_ex21\_continue.html

# 자바스크립트 함수

```
function ftn() {
```

```
    ...
```

```
}
```

# 함수 (function)

34

- 함수란?
  - ▣ 목적을 가지고 작성된 코드 블록
  - ▣ 데이터 전달받아 처리한 후 결과를 돌려주는 코드 블록
- 함수 개념



# 함수의 구성과 호출

35

## □ 함수의 구성

```
function 함수이름(arg1, arg2,..., argn) {  
    ...프로그램 코드...  
    결과를 리턴하는 return 문  
}
```

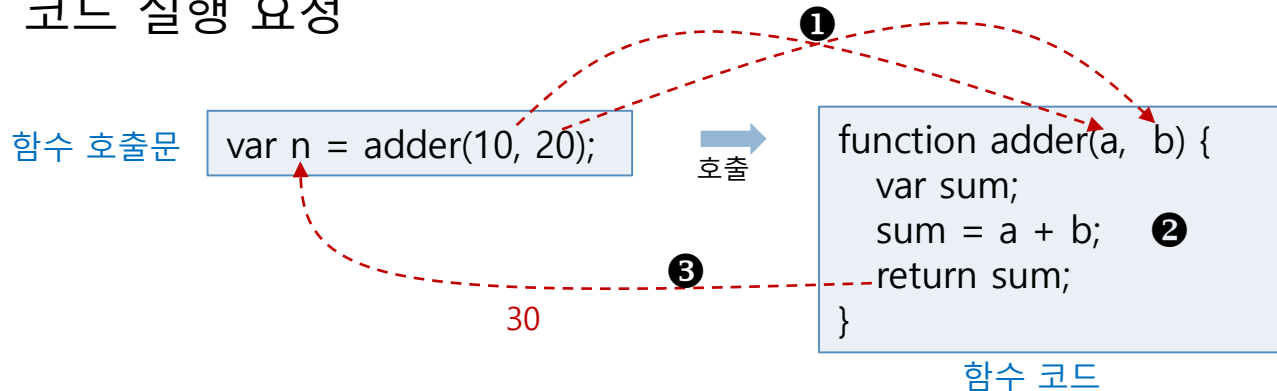
함수 선언      함수 이름      매개 변수

```
function adder ( a, b ) {  
    var sum;  
    sum = a + b;  
    return sum; // 덧셈 합 리턴  
}
```

반환 키워드      반환 값

## □ 함수 호출

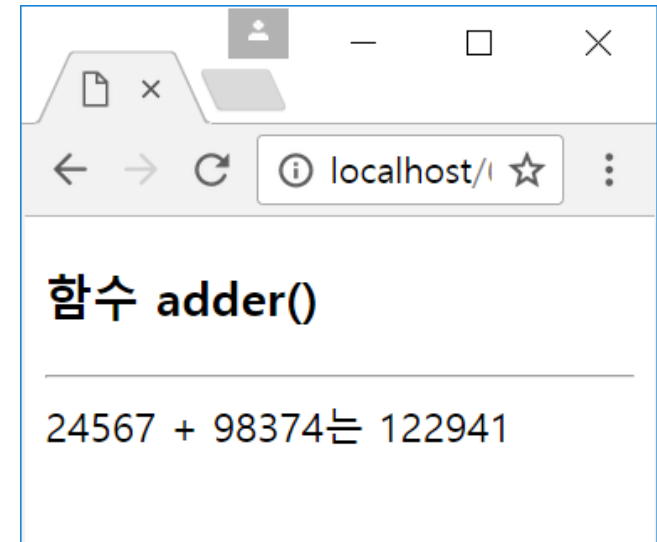
### ▣ 함수의 코드 실행 요청



# 예제 6-22 adder() 함수 작성 및 호출

36

```
<!DOCTYPE html>
<html>
<head>
<title>함수</title>
<script>
function adder(a, b) { // 함수 작성
    var sum;
    sum = a + b;
    return sum;
}
</script>
</head>
<body>
<h3>함수 adder()</h3>
<hr>
<script>
    var n = adder(24567, 98374); // 함수 호출
    document.write("24567 + 98374는 " + n + "<br>");
</script>
</body>
</html>
```



wk10js\_ex22\_function.html



# 자바스크립트에서 제공하는 전역 함수

37

## □ 대표적인 자바스크립트 내장 함수 (built-in functions)

### ▣ **eval()** 함수

예) `var res = eval("2*4+ 4*6");` // res는 32

### ▣ **parseInt()** 함수 // **parseFloat()**

예) `var i = parseInt("32");` // "32"를 10진수로 변환, 정수 32 리턴

`var n = parseInt("0x32");` // "0x32"를 16진수로 해석, 정수 50 리턴

### ▣ **isNaN()** 함수

예) `isNaN(32)` // false 리턴

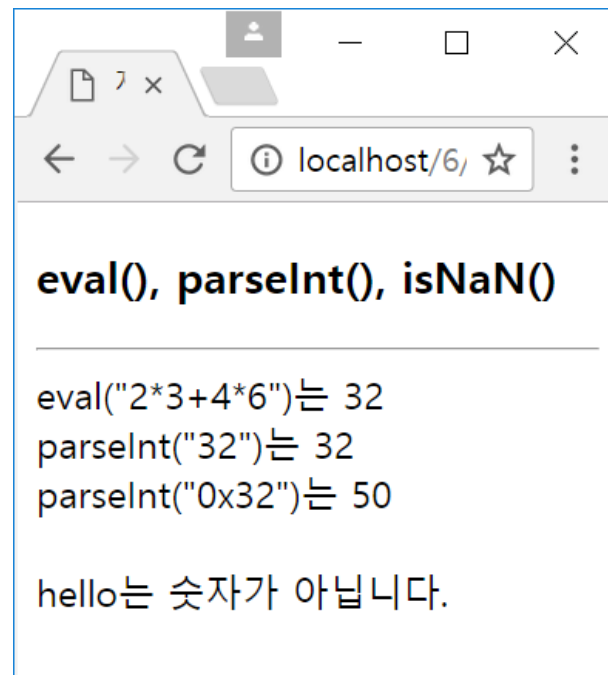
전역 함수명	설명
<code>eval(exp)</code>	exp의 자바스크립트 식을 계산하고 결과 리턴
<code>parseInt(str)</code>	str 문자열을 10진 정수로 변환하여 리턴
<code>parseInt(str, radix)</code>	str 문자열을 radix 진수로 해석하고, 10진 정수로 바꾸어 리턴
<code>parseFloat(str)</code>	str 문자열을 실수로 바꾸어 리턴
<code>isFinite(value)</code>	value가 숫자이면 true 리턴
<code>isNaN(value)</code>	value가 숫자가 아니면 true 리턴

# 예제 6-23 **eval(), parseInt(), isNaN()**

38

```
<!DOCTYPE html>
<html>
<head>
<title>자바스크립트 전역함수</title>
<script>
function evalParseIntIsNaN() {
    var res = eval("2*3+4*6"); // res는 32
    document.write("eval(W"2*3+4*6W")는 32<br>");
    var m = parseInt("32");
    document.write("parseInt(W"32W")는 " + m + "<br>");
    var n = parseInt("0x32");
    document.write("parseInt(W"0x32W")는 " + n + "<br><br>");
    var o = parseInt("17", 8);
    document.write("parseInt(W"17W", 8)는 " + o + "<br><br>");

    // "hello"는 정수로 변환할 수 없으므로 parseInt("hello")는 NaN 리턴
    n = parseInt("hello");
    if(isNaN(n)) // true
        document.write("hello는 숫자가 아닙니다.");
}
</script>
</head>
<body>
<h3>eval(), parseInt(), isNaN()</h3>
<hr>
<script>
    evalParseIntIsNaN();
</script>
</body>
</html>
```



# 예제 6-24 구구단 출력 함수 만들기

39

```
<!DOCTYPE html>
<html> <head> <title> 함수 만들기 </title>
<script>
function gugudan(n) { // 함수 작성
    var m = parseInt(n); // 문자열 n을 숫자로 바꿈
    if(isNaN(m) || m < 1 || m > 9) {
        alert("잘못입력하셨습니다.");
        return;
    }
    for(var i=1; i<=9; i++) { // i는 1~9까지 반복
        document.write(m + "x" + i + "=" + m*i + "<br>");
    }
}
</script>
</head>
<body>
<h3>구구단 출력 함수 만들기</h3>
<hr>
<script>
    var n = prompt("구구단 몇 단을 원하세요", ""); // n은 문자열
    gugudan(n); // 함수 호출
</script>
</body>
</html>
```

n이 1~9사이의 숫자가 아닌 경우 처리

localhost 내용:

구구단 몇 단을 원하세요

6

확인

취소

## 구구단 출력 함수 만들기

6x1=6  
6x2=12  
6x3=18  
6x4=24  
6x5=30  
6x6=36  
6x7=42  
6x8=48  
6x9=54

wk10js\_ex24\_gugudan.html

# 예제 섭씨-화씨 온도 변환앱

40

```
<html>
<head>
  <title>온도 변환기</title>
</head>

<body>
  <h3>온도 변환</h3>
  <hr>
  <table border="3">
    <tr>
      <td>섭씨온도</td>
      <td>화씨온도</td>
    </tr>
    <script>
      for (celsius = 0; celsius <= 10; celsius = celsius + 1) {
        document.write("<tr><td>" + celsius + "</td><td>"
          + ((celsius * 9.0 / 5) + 32) + "</td></tr>");
      }
    </script>
  </table>
</body>
</html>
```

$$F = (9/5) * C + 32$$

섭씨온도	화씨온도
0	32
5	41
10	50
15	59
20	68
25	77
30	86
35	95
40	104

# 예제 섭씨-화씨 온도 변환앱 (DIY-1)

41

섭씨-화씨 온도 변환  
함수를 만들어서  
결과를 출력하시오.

```
function c2f(c) {
```

```
}
```

**Javascript :**

**F <-> C Temperture Converter**

$$F = (9/5)*C + 32$$

섭씨온도(C)	화씨온도(F)
0	32
5	41
10	50
15	59
20	68
25	77
30	86
35	95
40	104
45	113
50	122

Save [hmx\\_x\\_temperature.html](#)

# JS-project 2: JS fancy calculator

42

## 간단한 계산기

첫번째 정수:

두번째 정수:

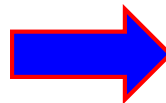
계산 결과:



## Javascript : Calculator

display

CSS3



## JS Calculator

COMSL, HM00

# Project 2: JS calculator

43

```
<html>
<head>

  <script>
    var expression="";

    function add(character) {
      expression = expression + character;

      document.getElementById("display").value = expression;
    }

    function compute() {
      document.getElementById("display").value = eval(expression);
    }

    function clearDisplay() {
      expression = "";
      document.getElementById("display").value = "0";
    }

  </script>
```

# Project 2: JS calculator

44

```
<form>
  display <input id="display" value="0" size="20">
  <br>
  <input type="button" value=" 7  " onclick="add('7')">
  <input type="button" value=" 8  " onclick="add('8')">
  <input type="button" value=" 9  " onclick="add('9')">
  <input type="button" value=" /  " onclick="add('/')">
  <br>
  <input type="button" value=" 4  " onclick="add('4')">
  <input type="button" value=" 5  " onclick="add('5')">
  <input type="button" value=" 6  " onclick="add('6')">
  <input type="button" value=" *  " onclick="add('*')">
  <br>
  <input type="button" value=" 1  " onclick="add('1')">
  <input type="button" value=" 2  " onclick="add('2')">
  <input type="button" value=" 3  " onclick="add('3')">
  <input type="button" value=" -  " onclick="add('.')">
  <br>
  <input type="button" value=" 0  " onclick="add('0')">
  <input type="button" value=" +  " onclick="add('+')">
  <br>
  <input type="button" value=" Clear " onclick="clearDisplay()">
  <input type="button" value=" Enter  " onclick="compute()">
</form>
```



# Project 2: JS calculator

45

**JS calculator**

<input type="button" value="7"/>	<input type="button" value="8"/>	<input type="button" value="9"/>	<input type="button" value="/"/>
<input type="button" value="4"/>	<input type="button" value="5"/>	<input type="button" value="6"/>	<input type="button" value="*"/>
<input type="button" value="1"/>	<input type="button" value="2"/>	<input type="button" value="3"/>	<input type="button" value="-"/>
<input type="button" value="0"/>		<input type="button" value="."/>	<input type="button" value="+"/>
<input type="button" value="Clear"/>		<input type="button" value="Enter"/>	

**COMSI, HMnn**

tip :



&larr ;

```
<div align="center">
  <h2>HMnn Calculator</h2>
  <form>
    <input id="display" value="0" size="18">
    <input type="button" value=" &larr; " onclick="del()">
    <br>
    <input type="button" value="7" " onclick="add('7')">
    <input type="button" value="8" " onclick="add('8')">
    <input type="button" value="9" " onclick="add('9')">
    <input type="button" value="/" " onclick="add('/')">
    <br>
    <input type="button" value="4" " onclick="add('4')">
    <input type="button" value="5" " onclick="add('5')">
    <input type="button" value="6" " onclick="add('6')">
    <input type="button" value="*" " onclick="add('*')">
    <br>
    <input type="button" value="1" " onclick="add('1')">
    <input type="button" value="2" " onclick="add('2')">
    <input type="button" value="3" " onclick="add('3')">
    <input type="button" value="-" " onclick="add('-')">
    <br>
    <input type="button" value="0" " onclick="add('0')">
    <input type="button" value="." " onclick="add('.')">
    <input type="button" value="+" " onclick="add('+')">
    <br>
    <input type="button" value="Clear" " onclick="clearDisplay()">
    <input type="button" value="Enter" " onclick="compute()">
  </form>
  <h4>COMSI, HM00</h4>
</div>
```

# Project 2: JS calculator

46

```
function add(character) {  
    expression = expression + character;  
    document.getElementById("display").value = expression;  
}  
function del() {  
    expression = expression.substring(0, expression.length - 1);  
    document.getElementById("display").value = expression;  
}  
function compute() {  
    var expression_num = parseFloat(eval(expression));  
    if (isInteger(expression_num) == true) {  
        expression = expression_num;  
    } else {  
        expression = expression_num.toFixed(2);  
    }  
    document.getElementById("display").value = expression;  
}  
function clearDisplay() {  
    expression = "";  
    document.getElementById("display").value = "0";  
}  
function isInteger(x) {  
    return Math.floor(x) === x;  
}
```

# Project 2: Design JS calculator - 1

47

Change style !

```
<style>
  input {
    padding: 8px;
    display: inline;
    margin-top: 4px;
    font-size: 20px;
  }
</style>
```

**JS calculator**

←

789/

456\*

123-

0. +

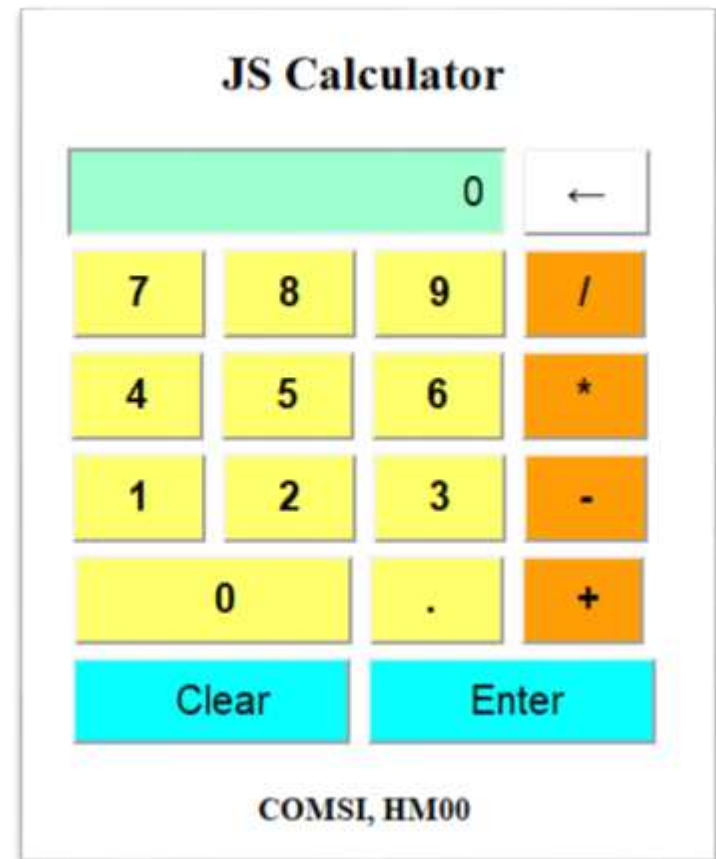
ClearEnter

COMSI, HMnn

# Project 2: Design JS calculator - 2

48

```
input {
  padding: 8px;
  display: inline;
  margin-top: 4px;
  font-size: 20px;
  background-color: #ffffff;
}
input#display {
  color: black;
  background-color: #99ffcc;
  font-size: 20px;
  text-align: right;
}
input.num {
  background-color: #ffff66;
  font-weight: bold;
}
input.op {
  background-color: #ff9900;
  font-weight: bold;
}
input#clear {
  background-color: #00ffff;
}
input#enter {
  background-color: #00ffff;
}
```



# JS-project-2. hmxx\_fancy\_calc.html

49

[JS project 2.] 자바스크립트를 이용하여 쓸만한 보기 좋은 계산기를 완성 하시오.

- ① 계산기 – 소스 수정 및 CSS 적용, 16진수 처리도 추가(bonus)
- ② 디자인 변경
- ③ 파일명: hmxx\_fancy\_calc.html
- ④ 가점: Javascript 프로그래밍 응용 능력.

# wk10-실습 : 결과를 나의 github에 올리기

50

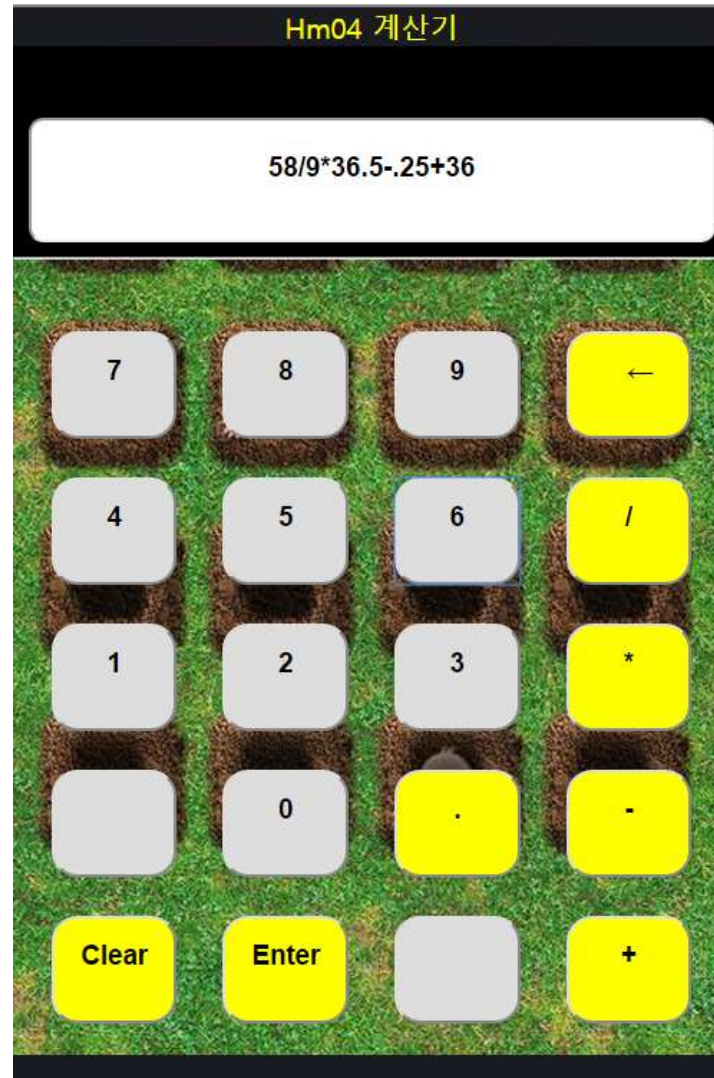
실습 결과를 github에 올립니다.

1. README.md에는 실습 결과 요약 추가 입력
2. hmxx\_temperature.html 완성
3. hmxx\_fancy\_calc.html 완성
4. "hmxx" repo 에 wk10 폴더 upload

단 업로드가 안될 경우, wk10.zip을 업로드  
그리고 집에서 wk10 폴더로 다시 업로드.

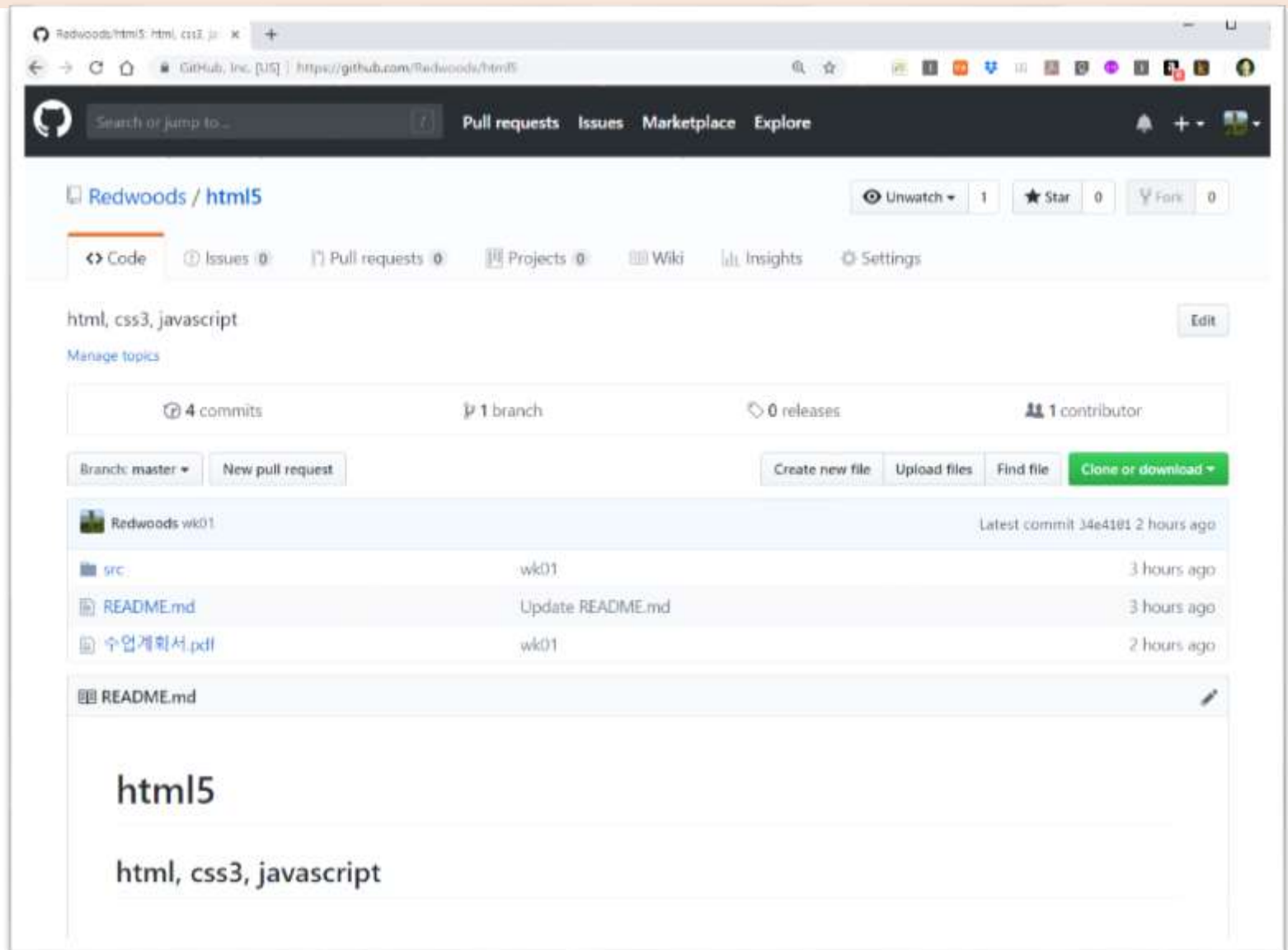
# Project 2: One of fancy JS calculator

51



# HTML5 : 강의자료실

## <https://github.com/redwoods/html5>





# 교재 WEB 강의 소개

← → ↻ ① webprogramming.co.kr ☆



HTML5+  
CSS3+  
Javascript

명품 웹 프로그래밍

Home Introduction Notice Board Support Code



HTML5 +  
CSS3 +  
Javascript

명품 웹 프로그래밍

HTML5로  
여러분의 무한한  
상상력을  
표현해 보세요!



Sir Tim Berners-Lee  
(1955.6.8 ~)

명품 웹 프로그래밍 소개  
"웹 프로그래밍을  
가장 쉽게 익힐 수 있는 책"



처음 웹 프로그래밍을 공부하는 입문자들도  
모든 주제를 직관적으로 이해하고  
빠르게 파악할 수 있습니다.

자세히보기 →

강력한 Q&A 피드백 제공  
"빠르고, 간결하고, 정확한  
저자의 직접적인 답변"



'이거 이해가 잘 안되는데.. 물어볼 사람도 없고..'  
더이상 고민하지 마세요.  
명품 웹 프로그래밍 홈페이지에서는  
누구나 저자가 직접 답변해주는  
Q&A 게시판들을 이용할 수 있습니다.

자세히보기 →

즉석 실행 가능한 예제 프로그램  
"백문이 불여일견, 백견이 불여일타(打)!"



코드뿐만 실행되어 있는 예제들.  
결과 화면이 있어도 이해가 잘 안되시죠?  
예제 소스를 바탕으로, 내림대로 수정한  
코드를 즉석으로 웹 페이지로  
변환해주는 예제 프로그램을 통해  
모든 코드를 빠르고 쉽게  
이해할 수 있습니다.

자세히보기 →

Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

53

<http://webprogramming.co.kr>

# 관련 WEB 강의 소개 – w3schools.com

The screenshot shows the w3schools.com website. The browser address bar displays "https://www.w3schools.com". The website has a green header with the logo "w3schools.com" and the tagline "THE WORLD'S LARGEST WEB DEVELOPER SITE". Below the header is a green navigation bar with links for "TUTORIALS", "REFERENCES", and "EXAMPLES".

On the left side, there is a sidebar menu with the following categories and links:

- HTML and CSS
  - Learn HTML
  - Learn CSS
  - Learn W3.CSS
  - Learn Colors
  - Learn Bootstrap
  - Learn Icons
  - Learn Graphics
  - Learn How To
- JavaScript
  - Learn JavaScript
  - Learn W3.JS
  - Learn JQuery
  - Learn JQueryMobile
  - Learn AppML
  - Learn AngularJS
  - Learn JSON
  - Learn AJAX
- Server Side
  - Learn SQL
  - Learn PHP
  - Learn ASP
- Web Building
  - Web Templates
  - Web Statistics
  - Web Certificates
- XML Tutorials
  - Learn XML
  - Learn XML AJAX
  - Learn XML DOM
  - Learn XML DTD
  - Learn XML Schema
  - Learn XSLT
  - Learn XPath
  - Learn XQuery

The main content area is divided into three sections:

- HTML**: The language for building web pages. It includes a "LEARN HTML" button and an "HTML REFERENCE" button. An "HTML Example:" box shows a code snippet for a basic HTML document structure.
- CSS**: The language for styling web pages. It includes a "LEARN CSS" button and a "CSS REFERENCE" button. A "CSS Example:" box shows a code snippet for styling a body and a div.
- JavaScript**: The language for programming web pages. It includes a "JavaScript Example:" box showing a code snippet for a function that gets an element by ID and changes its style.